Unsupervised Nearest Neighbors Clustering with Application to Hyperspectral Images

Claude Cariou, Kacem Chehdi

▶ **To cite this version:**

Claude Cariou, Kacem Chehdi. Unsupervised Nearest Neighbors Clustering with Application to Hyperspectral Images. IEEE Journal of Selected Topics in Signal Processing, 2015, 9 (6), pp.1105 - 1116. 10.1109/JSTSP.2015.2413371 . hal-01133648

HAL Id: hal-01133648
https://hal.science/hal-01133648

Submitted on 4 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Unsupervised Nearest Neighbors Clustering with Application to Hyperspectral Images

Claude Cariou, Kacem Chehdi

***Abstract*—We address the problem of unsupervised clustering of multidimensional data when the number of clusters is not known *a priori*. The proposed iterative approach is a stochastic extension of the kNN density-based clustering (KNNCLUST) method which randomly assigns objects to clusters by sampling a posterior class label distribution. In our approach, contextual class-conditional distributions are estimated based on a k nearest neighbors graph, and are iteratively modified to account for current cluster labeling. Posterior probabilities are also slightly reinforced to accelerate convergence to a stationary labeling. A stopping criterion based on the measure of clustering entropy is defined thanks to the Kozachenko-Leonenko differential entropy estimator, computed from current class-conditional entropies. One major advantage of our approach relies in its ability to provide an estimate of the number of clusters present in the data set. The application of our approach to the clustering of real hyperspectral image data is considered. Our algorithm is compared with other unsupervised clustering approaches, namely affinity propagation (AP), KNNCLUST and Non Parametric Stochastic Expectation Maximization (NPSEM), and is shown to improve the correct classification rate in most experiments.**

***Index Terms*—Data clustering, nearest neighbors, Bayes' decision rule, stochastic algorithm, differential entropy estimation, pixel classification, hyperspectral images.**

I. Introduction

Merging automatically objects having similar characteristics is a very important problem in various research fields such as computer vision, pattern recognition or information retrieval, when applied to medicine, genomics, chemistry, forensics, and more recently Big Data mining to cite a few [1], [2]. Despite several decades of research in this area, the task is still difficult because of the continual improvement of sensors technology and the increase of the size of data sets to analyze, both in the number of objects to handle (e.g. in very large size images or videos), and in the number of features that each object supports (e.g. DNA sequences or hyperspectral data).

Without any prior information about the data under study, the grouping of similar objects into coherent groups has to be done in an unsupervised way. This processing is called *unsupervised classification*, in contrast to *semi-supervised classification* which consists of grouping objects with the help of a small amount of *a priori* information on the data set, e.g. pairwise constraints (must-link / cannot-link) between objects, or a few number of class labels [3], and to *supervised classification* which requires a complete set of labeled data for training. In the literature, the term *clustering* generally refers to the family of unsupervised methods. The different groups of objects, characterized by features, are called clusters, which are formed of the closest objects, according to some specified distance between objects. Clustering methods themselves can be categorized into several families, comprising: centroid clustering (e.g. k-means [4], fuzzy c-means [5]); hierarchical clustering (e.g. based on minimum-variance [6] or single-link [7]); density-based clustering (e.g. DBSCAN [8], OPTICS [9], MeanShift [10]); clustering based on finite (EM [11], SEM/CEM [12], [13]) or countably infinite mixture resolving and Dirichlet process mixture models (DPMM) ([14], [15]); spectral clustering (e.g. normalized cuts [16] or kernel k-means [17]); and more recently information theoretic clustering ([18]–[24]), and clustering by Affinity Propagation (AP) [25].

Clustering methods can also be distinguished by the degree of randomness used to achieve the classification objective in the algorithms. Among the previously cited approaches, many are purely deterministic approaches to clustering (hierarchical methods, DBSCAN, OPTICS, MeanShift, AP to cite a few); on the other side are methods based on random partitioning or labeling (SEM, DPMM). Mixing cases comprise relaxation labeling methods, i.e. deterministic algorithms in which label probabilities are computed and iteratively updated until convergence to a stationary distribution [26], as well as basic centroid clustering approaches like k-means which require some random object labeling at initialization, though the body of the algorithm is purely deterministic.

In probabilistic methods, a standard approach uses the Bayesian paradigm, which classically requires a parametric modeling of class-conditional probability distributions. Each cluster is modeled by a multivariate distribution governed by a set of parameters, and the distribution of the objects in the data set is modeled as a linear combination of those conditional distributions [27]. A maximization of the likelihood function with respect to these parameters is then used to find the best parameters for each cluster. This maximization can be performed by the iterative EM algorithm [11]. The SEM algorithm [27], a stochastic version of the EM algorithm, can avoid the drawbacks of the EM algorithm such as slow convergence, convergence to local extrema of the likelihood function, or sensitivity to initial conditions [12]. Both the EM and SEM algorithms in their original design require the problem to be formulated as an *identifiable* mixture estimation problem, where the number of classes is known *a priori*, and the class-conditional distributions follow some parametric model (e.g. Gaussian distributions). However, a parametric modeling of

Copyright (c) 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Authors are with the TSI2M team, Institute of Electronics and Telecommunications of Rennes (IETR) - CNRS UMR 6164, University of Rennes 1 / Enssat, 6 rue de Kerampont, 22300 Lannion, France (e-mail: cariou@univ-rennes1.fr; chehdi@univ-rennes1.fr).

the conditional distributions is often difficult to assume in real cases because of their complex shapes, therefore justifying the need for non parametric approaches to data clustering.

Actually, a majority of these clustering methods require some prior knowledge about the data in the sense that the number of clusters to be found is explicitly provided by the user at the input of the algorithm. This is particularly true for centroid clustering, mixture resolving, and spectral clustering in their baseline implementation. In contrast, there is a more limited number of *fully unsupervised* clustering approaches which can *automatically* estimate the number of clusters, without any posterior selection of the number of clusters based on internal evaluation (like for instance Davies-Bouldin [28] or Dunn [29] indices). Among the latter are hierarchical agglomerative clustering methods, for which a stopping criterion still remains to be specified, as well as a number of density-based methods, including BIRCH [30] and DBSCAN [8]. Affinity Propagation (AP) [25] also has the capability to provide the optimal number of clusters at termination of the algorithm. There also exist a few number of elaborate centroid clustering methods which do not require the number of clusters, e.g. [31]. Another general framework issued from Bayesian nonparametric statistics and able to perform unsupervised clustering without knowing the number of cluster resides in Dirichlet process mixture models (DPMM) [15] [32] [14]. Stochastic processes like the Chinese restaurant process [33] and the stick-breaking process [34] aim to produce random partitions of a data set into clusters following a Dirichlet process mixture model.

In [35], a density-based clustering method named KNNCLUST was proposed. This method can be considered as an iterative unsupervised version of the kNNs (k nearest neighbors) algorithm which can automatically estimate the number of clusters, starting from a distinct cluster label for each object. At each iteration, for a given object, this method reassigns the cluster label of an object based on the current state of its kNNs and on its distance to them using Bayes' decision rule. KNNCLUST was successfully applied to multispectral images and is shown to outperform k-means and to provide results comparable to EM, with a relative insensitivity to the chosen number of nearest neighbors.

The present work is inspired from the Stochastic Expectation-Maximization (SEM) algorithm [27], a precursor of Markov chain Monte Carlo (MCMC) methods [36]. SEM is a stochastic extension of the Expectation-Maximization (EM) method [11] which tries to maximize the likelihood of a parametric model from realizations of the missing class membership data. Although SEM is primarily dedicated to parameter estimation, its use in clustering has been suggested as a stochastic version of the Classification EM algorithm (CEM) [13], which was recognized as a generalization of the k-means algorithm [37]. Both k-means and the baseline CEM share the same drawback of convergence to a local optimum or a saddle point of the likelihood function due to the deterministic nature of the class assignment step, whereas SEM is able to avoid such stationary points. In [38], a stochastic algorithm named NPSEM (non parametric SEM) was introduced, which proposes to replace, in SEM, the parametric conditional distributions by non-parametric estimates based on both kernel-based pairwise similarities between objects, and an estimation of class-conditional entropies, from which a posterior distribution (of the cluster assignment given an observation or object) is computed and used to sample the next labeling state. NPSEM was compared and found experimentally to be slightly superior to k-means, FCM [5], FCM-GK [39], EM with Gaussian conditional distributions [37], and KNNCLUST, on a limited set of data.

In this paper, we present a new fully unsupervised clustering technique which we name KSEM, standing for Kernel SEM. KSEM is in the spirit of NPSEM since it is intended to perform unsupervised clustering by iteratively reassigning objects to clusters, following a stochastic procedure. KSEM also provides an estimate of the number of clusters which, contrarily to NPSEM, and as an improvement of it, does not require any upper bound on the number of clusters, nor any minimum number of objects in a cluster. KSEM is an iterative procedure which produces at each step a random partition of the data set objects using local pseudo-posterior distributions, until a differential entropy-based criterion is met, revealing a stable partition.

The paper is organized as follows. Section II provides a presentation of our algorithm, followed by a discussion focusing its novelty with respect to NPSEM and KNNCLUST. In Section III we present experimental results focusing the unsupervised classification of pixels in hyperspectral images, in which we compare KSEM with three other clustering methods, namely AP, KNNCLUST and NPSEM. Finally, a conclusion of this work is given in Section IV.

II. Proposed Clustering Method

In this section, we describe the proposed clustering method, KSEM, point out several practical issues, and then discuss its relationships with the KNNCLUST [35] and NPSEM [38] algorithms.

A. Proposed method

Let $\mathbf{X}$ denote the original data set to cluster, $\mathbf{X} = \{\mathbf{x}_i\}, \mathbf{x}_i \in \mathbb{R}^n, i = 1, \ldots, N$, where $\mathbf{x}_i$ is called an object. $\mathbf{X}$ can be thought as a collection of N realizations of n-dimensional random vectors. Let us denote by C_i a discrete random variable (r.v.) corresponding to the class label held by object $\mathbf{x}_i$; let c_i be an outcome label sampled from some distribution on $C_i, i = 1, \ldots, N$. The objective of the proposed method is to assign a label to each object according to a modified maximum likelihood criterion, following a random sampling scheme similarly as NPSEM [38]. From a general viewpoint, our method works by iteratively partitioning the objects into clusters by sampling from local posterior distributions $p(C_i|\mathbf{x}_i; \{\mathbf{x}_j, c_j\}_{j \neq i}), i = 1, \ldots, N$.

This general formulation raises several questions, among which (*i*) what is the set of objects and corresponding labels $\{\mathbf{x}_j, c_j\}$ to consider for some $\mathbf{x}_i$? (*ii*) how can be estimated the posterior probability distribution used to get the current cluster label of an object? and (*iii*) what type of sampling scheme can be set up in the algorithm?

The first issue is related to the overall support model for label assignment. For instance, a model-based approach through Conditional Random Fields (CRF) [40] can be used to set up a probabilistic dependency restricted to some prescribed neighborhood of $\mathbf{x}_i$ with respect to a graph. In KSEM, we propose to assign a label to every pixel, based on the labels of its k nearest neighbors (kNNs) in the representation space, hence to use a nearest neighbors graph model. Let k be the (fixed) number of nearest neighbors , $\kappa(i)$ be the set of indices of the kNNs of $\mathbf{x}_i$, i.e. such that $\mathbf{x}_j$ is a kNN of $\mathbf{x}_i$ $\forall j \in \kappa(i)$; let $\mathbf{c} = [c_1, \ldots, c_N]^T$ be the vector of cluster labels, where $c_i \in \Omega \subset \{1, \ldots, N\}$, Ω being the set of distinct labels taken by the set of all objects; and let $\Omega(i) = \{c_j | j \in \kappa(i)\}$ be the set of distinct labels taken by the kNNs of $\mathbf{x}_i$. Since duplicate labels can be held by the kNNs of an object, we have $|\Omega(i)| \leq k$, and consequently $|\Omega| \leq N$.

Regarding the second issue, the local posterior label distribution in KSEM can be modelled primarily as:

$$\hat{p}(C_i = c_\ell | \mathbf{x}_i; \{\mathbf{x}_j, c_j\}_{j \in \kappa(i)}) \propto \sum_{j \in \kappa(i)} g(\mathbf{x}_j, \mathbf{x}_i)\ \delta_{c_j c_\ell} \qquad (1)$$

$\forall c_\ell \in \Omega(i),\ 1 \leq i \leq N$, where g is a (non negative) kernel function defined on $\mathbb{R}^n$, δ_{ij} is the Kronecker delta. Though many kernel functions can be used, including rectangular, triangular or Epanechnikov [35], we have restricted our study to the following Gaussian kernel:

$$g(\mathbf{x}, \mathbf{x}_i) = \frac{1}{\left(\sqrt{2\pi} d_{k,\kappa}(\mathbf{x}_i)\right)^n} \exp\left(-\frac{1}{2} \frac{\|\mathbf{x} - \mathbf{x}_i\|_2^2}{d_{k,\kappa}^2(\mathbf{x}_i)}\right) \quad , \qquad (2)$$

where $\mathbf{x} \in \mathbb{R}^n$, and $d_{k,\mathcal{S}}(\mathbf{x}_i)$ represents the distance from $\mathbf{x}_i$ to its kth NN within a subset $\mathcal{S}$ of objects. Clearly, $d_{k,\kappa}(\mathbf{x}_i)$ is the distance from $\mathbf{x}_i$ to its kth NN in $\kappa(i)$, and we have $d_{k,\kappa}(\mathbf{x}_i) = \max_{j \in \kappa(i)} ||\mathbf{x}_j - \mathbf{x}_i||$. Therefore, the kernel function is adapted in aperture to the local density around each $\mathbf{x}_i$. This modeling is similar to the one described in [35], but is simpler to implement thanks to rotational symmetry.

Once the above model of the label distribution is available, we can consider the third question mentioned above about the labeling procedure itself. In KNNCLUST [35], the labeling of object $\mathbf{x}_i$ is based on Bayes' decision rule, i.e. the label $c_\ell^\star$ maximizing Eq. (1) is chosen, and thus $c_i = c_\ell^\star$. Though this rule is very simple to understand and has a strong justification in decision theory, and despite its attractive deterministic nature, it is expected to suffer from the same drawback than the EM algorithm applied to augmented data for mixture estimation, i.e. of trapping the solution into a local optimum or a saddle point of the likelihood function.

In order to avoid this problem, we propose to replace the deterministic label selection procedure by a random sampling procedure in which the label of each object is drawn from the local posterior distribution (1). This distribution can be modified easily in order to get a trade-off between the purely stochastic (SEM-like) and the purely deterministic (CEM-like) approaches, as follows [41]:

$$\hat{p}_\alpha(C(i) = c_\ell | \mathbf{x}_i; \{\mathbf{x}_j, c_j\}_{j \in \kappa(i)}) = \frac{\left[\sum_{j \in \kappa(i)} g(\mathbf{x}_j, \mathbf{x}_i)\ \delta_{c_j c_\ell}\right]^\alpha}{\sum_{c_m \in \Omega(i)} \left[\sum_{j \in \kappa(i)} g(\mathbf{x}_j, \mathbf{x}_i)\ \delta_{c_j c_m}\right]^\alpha} \qquad (3)$$

$\forall c_\ell \in \Omega(i),\ 1 \leq i \leq N$, where $\alpha \in [1, +\infty[$ is a parameter controlling the degree of determinism in the construction of the pseudo-sample: $\alpha = 1$ corresponds to the SEM (stochastic) scheme, while $\alpha \to +\infty$ corresponds to the CEM (deterministic) scheme, leading to a labeling scheme which is similar to the k nearest neighbors decision rule, where distances between objects in the representation space are replaced by their mutual influence via kernel functions [41]. It can be mentioned that the CAEM algorithm [13] is a generalization of this principle which considers the exponent α as the inverse of a temperature in an annealing scheme. This parameter is also to some extent comparable to the fuzziness parameter used in the semi-supervised FCM algorithm.

B. Practical issues

Several practical issues must be discussed about the applicability of the proposed method to clustering objects in real data sets:

1) Stopping criterion: A general framework to deal with cluster consistency relies in information theory. Recently, information theoretic (IT) clustering [18]–[24] has become an attractive approach due to its ability to cope with arbitrary shaped clusters. IT clustering is based on information theoretic measures of the cluster consistency, using cluster-conditional entropy as the within-cluster validation criterion, or the mutual information between the given labels and the clustered data. Several schemes have been proposed to optimize these criteria, including minimum spanning trees [22] and convex optimization [23]. It can be noticed that these methods all have the problem of requiring in advance the knowledge of the number of clusters to discover. This was also the case of former entropy-based clustering algorithms used in language modelling [42], [43]. In [44], Kozachenko and Leonenko have proposed an unbiased estimator of the differential entropy of a random vector from a set of observations in $\mathbb{R}^n$, as a function of averaged log-distances between objects and their 1-NNs. This estimator has been generalized to the case of log-distances to kNNs in [45], [46] and its unbiasedness and consistency were proved. More precisely, letting $\mathcal{S}_{c_\ell} = \{\mathbf{x}_i \in \mathbf{X} | c_i = c_\ell\}$, the Kozachenko-Leonenko conditional differential entropy estimate writes:

$$\hat{h}(\mathbf{X}|c_\ell) = \frac{n}{N_\ell} \sum_{\mathbf{x}_i \in \mathcal{S}_{c_\ell}} \ln d_{k,\mathcal{S}_{c_\ell}}(\mathbf{x}_i) + \ln(N_\ell - 1) - \psi(k) + \ln V_n \qquad (4)$$

$\forall c_\ell \in \Omega$, where $N_\ell = |\mathcal{S}_{c_\ell}|$, $\psi(k) = \Gamma'(k)/\Gamma(k)$ is the digamma function, $\Gamma(k)$ is the gamma function and $V_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)}$ is the volume of the unit ball in $\mathbb{R}^n$. An overall clustering entropy measure can be obtained from conditional entropies (4) as:

$$\hat{h}(\mathbf{X}|\mathbf{c}) = \frac{1}{N} \sum_{c_\ell \in \Omega} N_\ell \; \hat{h}(\mathbf{X}|c_\ell) \quad . \tag{5}$$

This measure can be used as a stopping criterion during the iterations quite naturally. In fact, under the assumption of a fixed number of clusters, $\hat{h}(\mathbf{X}|\mathbf{c})$ is expected to decrease until a valid partitioning of the data set is found. However, the situation is different when the number of clusters is not given in advance but is decreasing during the course of the labeling procedure. Clearly, since objects are aggregated into previously formed clusters during the iterations, the individual class-conditional entropies can only increase, and so does the conditional entropy (5). However, when convergence is achieved, this measure reaches an upper limit, and therefore a stopping criterion can be set up from its relative magnitude variation $\Delta_h = |\hat{h}(\mathbf{X}|\mathbf{c}^{(t)}) - \hat{h}(\mathbf{X}|\mathbf{c}^{(t-1)})|/\hat{h}(\mathbf{X}|\mathbf{c}^{(t-1)})$, where $\mathbf{c}^{(t)}$ is the vector of cluster labels at iteration t. In our experiments, we have used the stopping criterion $\Delta_h < 10^{-4}$. It is interesting to mention that (5) has been used as a basis of several IT clustering methods under different schemes (see [20], [21], [23] for some recent works), and was proved in [23] to overcome the main flaw of the mutual information clustering criterion $I(\mathbf{X}\,;\mathbf{c})$ which has tendency to ignore the data set structure.

2) Choice of k: The number of NNs k involved in the computation of the posterior distribution is actually the key parameter of the proposed method, similarly to KNNCLUST. The influence of this parameter can be easily anticipated; indeed, increasing k will tend to promote a few number of labels propagating on the kNN graph, whereas decreasing k will tend to produce a high number of clusters since label sampling remains local. This issue will be further investigated in Section III.

3) Choice of α: Recall that setting $\alpha = 1$ is equivalent to performing the random label assignment following the original posterior local distribution derived from Bayes' decision rule. However, it can be observed that this setting generally slows down the convergence to a final clustering solution. The probability reinforcement parameter $\alpha = 1.2$ was found to be a good trade-off between the randomness of the labeling scheme and the convergence speed, and was chosen in all our experiments.

4) Complexity: In NPSEM [41], each object of the data set is involved in the labeling decision rule of one particular object. Though such a rule remains tractable for small size data sets (a few thousands objects), it becomes computationally infeasible for large data sets (several millions of objects or more). Such a situation is also encountered in other unsupervised clustering methods such as AP [25] for which pairwise similarities between objects are required, making the algorithm quadratic in the number of objects N. Therefore performing the labeling decisions dependent on a reduced set of neighboring objects (given by the kNN search), is highly desirable. Actually, the complexity of a single iteration of KSEM (as well as KNNCLUST) is majored by $\mathcal{O}(k^2 N)$ at the beginning of the algorithm (since k different labels are assigned to the kNNs of an object), and approximately minored by $\mathcal{O}(k.NC.N)$, where NC is the number of final clusters.

5) Convergence: Formal convergence properties of KSEM are not easy to establish and will not be investigated herein. Nevertheless, the vector of labels drawn from posterior distributions can be seen as issued from an aperiodic inhomogeneous Markov chain with (local) absorbing states due to the removal of labels having lowest probabilities during the iterations. Indeed, labels which are not drawn at all for the whole set of objects simply disappear from Ω, therefore reducing the state space. However, convergence to a stable, non trivial clustering result (i.e. different from a single final cluster) has been experimentally observed in all encountered cases. It is important to notice that the clustering result is independent of any initial labeling since each object is assigned a single, unique label at the beginning of the algorithm. We also found experimentally (see below in Section III) that the number of iterations is significantly higher than for KNNCLUST (by a factor around 4), but only slightly higher than for NPSEM. This fact can be explained by the stochastic nature of KSEM and NPSEM versus KNNCLUST.

C. Application to images

Despite the reduction in complexity brought by the kNN search, the case of image segmentation by unsupervised clustering of pixels with KSEM remains computationally difficult; indeed, the search for objects' kNNs which must be performed (and stored) beforehand still remains quadratic in N (the number of pixels), which can severely lower its usage for large size images. In the particular domain of multivariate imagery (multispectral/hyperspectral), the objects of interest are primarily grouped upon their spectral information characteristics. To help the clustering of image pixels, one makes often use of the spatial information, and of the fact that two neighboring pixels are likely to belong to the same cluster [47]. In order to further reduce this complexity, we propose to limit the search of a pixel's kNNs to a subset of its *spatial* neighbors, selected via a predefined sampling pattern. The sampling pattern chosen here is non-uniform on the spatial image grid, with a higher sampling density in the vicinity of the query pixel, and a lower density as the spatial distance to the query pixel increases. Figure 1 shows the spatial sampling pattern which was used in our experiments. This pattern has a local sampling density inversely proportional to the distance from the central (query) point. Obviously many other sampling schemes may apply, however we have not investigated this issue in the present work. If M is the number of candidate sampled pixels ($M = 185$ in Figure 1), then the kNN search procedure has complexity $\mathcal{O}(MN)$, which is dramatically lower than a full search over the entire image, even for small size images.

The pseudo-code of the KSEM algorithm is given in Algorithm 1. Since only a limited number of pairwise distances (and not the data values themselves) are required to compute the posterior distributions, these are first stored into a table which can be easily accessed during the iterations.

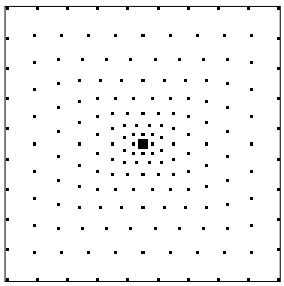


Fig. 1. Example of spatial sampling pattern used to limit the k-NN search in images (sliding window size 91×91, 185 sampling pixels shown in black). The central point of the window corresponds to the query pixel in the kNN search, whereas the remaining points correspond to the set of potential k-NN pixels.

Algorithm 1 KSEM algorithm

Input:

$\mathbf{X} = \{\mathbf{x}_i\}, \mathbf{x}_i \in \mathbb{R}^n, i = 1, \ldots, N$;
The number of NNs k;
The probability reinforcement parameter α;
The stopping constant ϵ;

Output: The vector of final cluster labels $\mathbf{c} = [c_1, \ldots, c_N]^T$;

1) Compute the $N \times k$ matrix D containing the distances of each object to its NNs up to the k-th NN;
(in case of image data, use the sampling scheme described in Section II-C).
2) Initialize labels $\mathbf{c}^{(0)} = \left[c_1^{(0)}, \ldots, c_N^{(0)}\right]^T = [1, 2, \ldots, N]^T$; Let $\Omega^{(0)} = \{1, 2, \ldots, N\}$;
3) Iterations:
$t = 0$;
$\Delta_h = 1.0$;
while $\Delta_h \geq \epsilon$ **do**
  Compute the overall class-conditional entropy $\hat{h}(\mathbf{X}|\mathbf{c})$ (Eq. (5)) using the log of distances stored in D;
  for $i = 1 : N$ **do**
    for all $c_\ell \in \Omega^{(t)}(i)$ **do**
      Compute the posterior probability:
      $\hat{p}_\alpha(C_i = c_\ell | \mathbf{x}_i; \{\mathbf{x}_j, c_j^{(t)}\}_{j \in \kappa(i)})$; (Eq. (3))
    end for
    Draw a new label:
    $c_i^{(t+1)} \sim \hat{p}_\alpha(C_i | \mathbf{x}_i; \{\mathbf{x}_j, c_j^{(t)}\}_{j \in \kappa(i)})$;
  end for
  $\mathbf{c}^{(t+1)} = \left[c_1^{(t+1)}, \ldots, c_N^{(t+1)}\right]^T$;
  Update $\Omega^{(t+1)}$ by counting the remaining distinct labels;
  $\Delta_h = \frac{|(\hat{h}(\mathbf{X}|\mathbf{c}^{(t+1)}) - \hat{h}(\mathbf{X}|\mathbf{c}^{(t)}))|}{\hat{h}(\mathbf{X}|\mathbf{c}^{(t)})}$;
  $t \leftarrow t + 1$;
end while

D. Discussion

KSEM brings some important advantages with respect to either KNNCLUST and NPSEM which it is inspired from.

First, the key idea of KSEM is to avoid the limitations of KNNCLUST due to crisp decisions taken at each iteration at the object level by allowing the current object label to be chosen among the set of labels of its kNNs. The random sampling procedure within KSEM clearly avoids the solution to be trapped in a local optimum of the likelihood $p(\mathbf{X}; \mathbf{c})$. As said above, this is a well known property of the SEM algorithm and its derivatives [13] in a parametric context. However, it must be noticed that KNNCLUST, although this is a deterministic algorithm in essence, still offers the possibility to produce different results from a unique initialization label state, by visiting the objects in random order rather than sequentially in turn. We have used this feature in the experimental study presented below, since it allows to compare the clustering performances on a statistical basis from results obtained by independent runs of the algorithm. Another difference of KSEM with respect to KNNCLUST relies in the specification of the kernel function (2). Indeed, for the Gaussian kernel proposed in [35], the volume of the bin around each object is adapted in scale along each dimension of the representation space to include its kNNs. Here, the kernel is rotationally symmetric and only dependent of the Euclidean distance $d_{k,\kappa}(\mathbf{x}_i)$. The motivation for using this particular kernel relies in a reduced computational load, and also to a lesser sensitivity to very close objects along a specific dimension, since $d_{k,\kappa}(\mathbf{x}_i)$ is very unlikely to be close to zero for moderate values of k (a few tens).

Second, NPSEM has several remaining problems, among which the choice of the upper bound of the number of classes which must be initialized by the user, and the fact that the distance separating all pairs of objects is required to compute the posterior label probability distribution. This latter requirement prohibits its use for large data sets (i.e. with a large number of objects, the dimension of the representation space being not an issue here) due to the quadratic complexity implied by the pairwise distance computation and storage. Contrarily, KSEM (*i*) does not need the initialization of an upper bound on the number of clusters, nor any minimum number of objects assigned to a cluster, and (*ii*) is based on a kNN graph, therefore requiring much less storage capability.

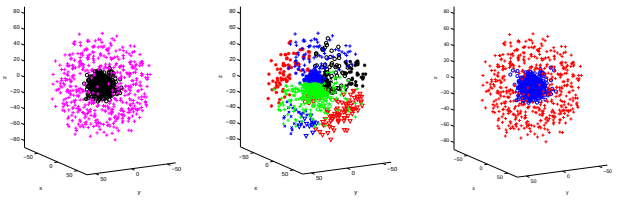
III. Experiments and Results

A. Synthetic data set

In order to demonstrate the validity of our approach in non-linear discriminant clustering, we first illustrate its application to a synthetic data set. In this example, 1000 3-D objects are generated randomly following two distribution models: the first one is a multivariate Gaussian distribution centered at the origin of coordinates, and with covariance matrix $\Sigma = 64\mathbf{I}$; the second one is a distribution surrounding the first one, specified by a radius from the origin following a normal univariate distribution $\mathcal{N}(50, 64)$. 500 objects are assigned to each one of the two distributions.

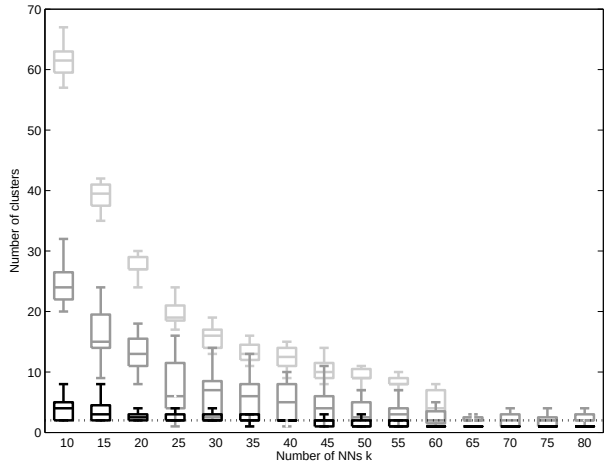
Figure 2 shows the data set with true labels, as well as the corresponding KNNCLUST and KSEM results. Using KSEM with $k = 30, \alpha = 1.2$, two clusters were found, and the overall classification error rate is 0.8% (8 pixels misclassified). This result compares well to the theoretical classification error rate of 0.86%. In comparison, KNNCLUST with the same number of NNs provided 15 classes, thus far from the true number of classes. To get a more precise idea of the clustering

stability of these algorithms, we show in Figure 3 compared box plots of the numbers of final clusters and associated classification rates obtained with KNNCLUST, NPSEM (with initial number of clusters $NC_{\max} = 100$) and KSEM. These results were obtained from 20 independent runs for each method. Concerning the number of clusters, one can see that the true number of cluster is in average better identified by KSEM in the range $10 \leq k \leq 40$, which is in agreement with higher corresponding average classification rates in this range.

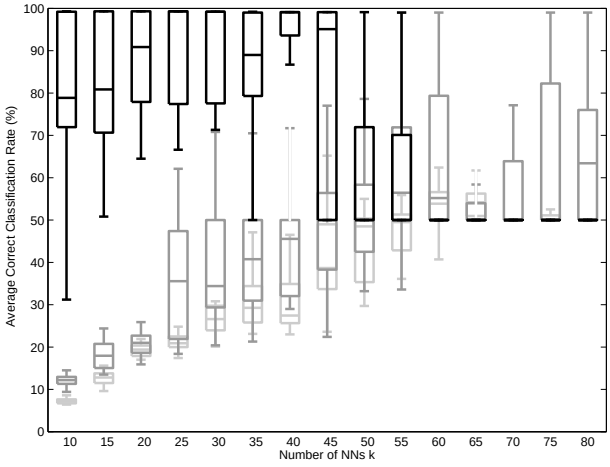


(a) (b) (c)

Fig. 2. Clustering of synthetic 3-D data. (a): Original data and corresponding true labels; (b): KNNCLUST result; (c): KSEM result.



(a)



(b)

Fig. 3. Comparison of the evolution of the number of clusters (a) and overall correct classification rates (b) given by KNNCLUST (light gray), NPSEM (gray) and KSEM (black). The horizontal dotted line in (a) represents the true number of classes $NC = 2$.

B. HSI Clustering: methodology

We provide now an experimental study of the performances of the proposed approach, focusing the segmentation of hyperspectral images (HSIs) by unsupervised clustering for remote sensing applications. Airborne and spatial hyperspectral imaging has received much attention since two decades both from the end-users due to the richness of information that HSIs carry, and by the community of image and signal processing and analysis experts due to the diversity and the complexity of the problems that multivariate image analysis poses to achieve end-user objectives in terms of classification, spectral unmixing, or anomaly detection. In a HSI, the objects $\{\mathbf{x}_i\}, 1 \leq i \leq N$, are associated to image pixels, and the entries of $\mathbf{x}_i$ are (positive) radiance or reflectance values collected at n spectral wavelengths in the visible (VIS) to near-infrared (NIR) or short-wave infrared (SWIR) range. HSIs allow to accurately characterize and distinguish natural and man-made materials through absorption/emission, narrow/wide spectral bands. It is worth mentioning that our experiments were performed without prior band selection or feature extraction. Even the noisy, low average reflectance spectral bands often present in the HSI at some absorption bands were preserved in the input data set.

1) Selected methods for comparison: The experiments were designed to assess the performances of the proposed method in comparison with similar fully unsupervised clustering approaches, i.e. methods which do not require any prior information about the data objects to be classified and particularly the true number of clusters to be discovered. Among the variety of approaches in the domain, we have selected Affinity Propagation (AP) [25], kNN density-based clustering KNNCLUST [35], and Non parametric SEM (NPSEM) [41]. The choice of these particular methods was motivated by the fact that they all share the same initial conditions than KSEM, since a unique label is given to each object of the data set at the beginning of the algorithm. This allows to compare the four methods on the same basis, and makes initial conditions a non issue in this comparison. The DPMM approach, which requires a greater number of prior parameters than the above methods (among which a concentration parameter and an upper bound on the number of clusters), was not included in this study.

Each of the selected methods requires a couple of parameters that can be tuned to provide more or less accurate clustering results. These parameters generally influence the number of clusters at the output of the algorithm. Concerning AP, it is recognized in the literature that this method is sensitive to the choice of the *preference* parameter, which governs the way an object considers itself a better exemplar for a cluster than any other object. It has been shown recently in [48] that the rule of thumb of choosing the median value of pairwise similarities as the preference parameter p for hyperspectral data generally leads to over-clustering, and that a better estimate of p regarding the final number of clusters is closer to the minimum of the similarities $s(i,j) = -||\mathbf{x}_i - \mathbf{x}_j||^2$ than to their median value as often recommended [25]. Thus, we have chosen the parametrization $p = \xi . \min_{i,j} s(i,j)$, leaving ξ as the only parameter for AP. Regarding NPSEM,

one important parameter is the upper bound on the number of clusters $NC_{\max}$. In all our experiments, we have fixed $NC_{\max} = 100$. Concerning KNNCLUST and KSEM the only tunable parameter is k, the number of NNs, the probability reinforcement parameter α being fixed to 1.2 throughout the experiments.

Another point relates to the complexity of the chosen methods when tackling data sets comprising a high number of objects, which is the case for the chosen hyperspectral images (see below). Since AP is complex in $\mathcal{O}(N^2)$ which is intractable for high N, we randomly selected throughout the HSI a subset of N' objects (or pixels), having the maximal size allowed by the computer environment, and AP was run on this subset. The centroids of the resulting clusters were computed and used to cluster the remaining $N - N'$ objects based on a minimum distance rule. The cardinality of the subset was fixed to $N' = 12962$ whatever the HSI processed. Concerning KNNCLUST and KSEM, the spatial sampling scheme described in Section II-C was used instead, allowing their comparison under identical conditions. NPSEM, which originally was proposed in an exhaustive pairwise distance setting, was also adapted to the kNN graph setting by removing the most distant pairs of objects in the computation of posterior distributions. Note that a transposition of this principle has been tried for AP but could not yield satisfactory results, providing a much higher number of clusters than expected. This result is probably due to the kNN graph structure for which message passing remains local and is barely influenced by messages passed outside the scope of each object's kNNs.

2) Clustering assessment: In order to assess the clustering results, we have chosen HSIs with available ground truth data. To obtain clustering performance indices when the number of clusters found is greater than the number of known ground truth classes, it is necessary to find the best match between the cluster labels and the ground truth labels. For this we first construct the confusion matrix (CM) of size $NC_{\text{gt}} \times NC_{\text{clus}}$, where NC_{gt} is the number of ground truth classes, and NC_{clus} is the number of output clusters. This CM is then augmented with $NC_{\text{clus}} - NC_{\text{gt}}$ zeroed rows, and the best class-cluster assignment is sought thanks to the Hungarian algorithm [49], and applied to permute the columns of the augmented CM, providing a new CM with maximal trace. Therefore, classical CM-based clustering performance indices can be accessed such as the overall correct classification rate (OCCR), i.e. the trace of the new CM divided by the total number of pixels, as well as the class-specific correct classification rate (CSCCR), i.e. for each ground truth class the number of pixels correctly predicted divided by the number of pixels belonging to that ground truth class, and the average correct classification rate (ACCR), i.e. the average of the CSCCRs over the number of ground truth classes. The Cluster Purity and Normalized Mutual Information (NMI) indices [50] have also been used for comparison. These indices both have maximal unity value for an error-free clustering result.

Since the number of ground truth pixels is often small with respect to the spatial size of the images to analyze, it can be interesting to assess the quality of each method by counting the number of clusters found within the known ground truth pixels only; this one is expected to be close to the number of known classes for a good clustering result. This is why we have also added for each method the number of clusters found within the ground truth pixels in the Tables providing the classification results.

Finally, in order to statistically assess the performance indices, their averages and standard deviations were computed from 10 or 20 independent runs for each method, depending on the HSI under study.

C. HSI Clustering: results

1) AVIRIS - Salinas: This HSI was acquired by the AVIRIS sensor over agricultural crops in Salinas Valley, California, on October 8, 1998. Its size is 512 lines by 217 samples, and 220 spectral bands are available. The ground resolution is around 4 meters. The ground truth map reports 16 vegetation classes, some of them representing several stages of growth (lettuce) or different agricultural practices (vineyard) [51]. Figure 4 shows a color composite of the *Salinas* scene, and the corresponding ground truth map.

We first studied the influence of the number of neighbors k on the classification accuracy provided by KNNCLUST, NPSEM and KSEM. For this, we performed 20 independent runs of the three methods for values of k in the range from 10 to 80. Figure 5 displays the box plots of the ACCR versus k for the three methods. One can observe dissimilarities between the three methods in terms of accuracy as k evolves. Firstly, ACCR maxima are obtained for different values of k, and the optimum is found for $k \approx 50$ with KNNCLUST, whereas $k \approx 20$ is the optimum for NPSEM and $k \approx 40$ for KSEM. Secondly, this study provides a comparison of the efficiency of the three clustering methods in terms of ACCR, and shows that KSEM can outperform the two other methods for some adequate range of k, here $k \leq 40$. Contrarily, for $k > 40$, KNNCLUST provides the best results among the three methods, though with significantly decreasing accuracy as k increases. Thirdly, one can observe the lower dispersion of ACCR around their average values with KSEM for $k \leq 40$ compared to the other methods, which denotes a higher stability of our approach. This is particularly true for k around 30-40, i.e. in its optimal range. Therefore a careful choice of k must be made before using each method.

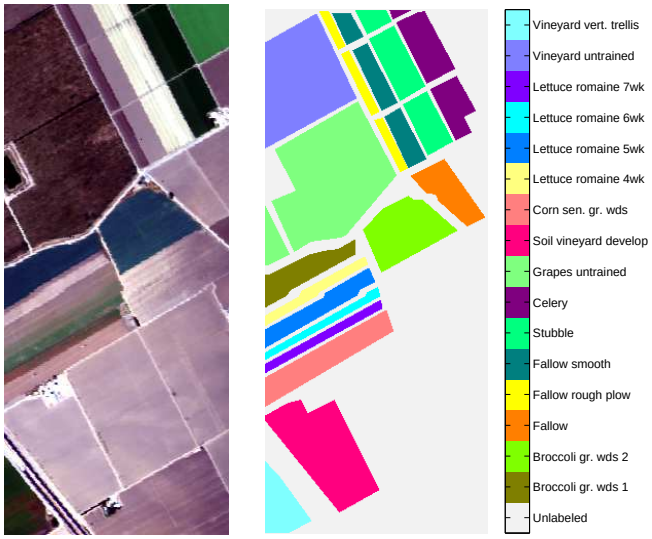
Table I reports a detailed comparison of clustering results using the optimal values of k issued from this analysis, i.e. $k = 50$ for KNNCLUST, $k = 20$ for NPSEM ($NC_{\max} = 100$), $k = 40$ for KSEM. Results provided by AP are also included. Yet, 11 out of the 16 classes were better identified by KSEM, giving 82.44% average ACCR, and 79.17% average OCCR over the 20 runs. Also, the median number of clusters found within the labeled data, 18, is close to the actual number of classes. Though its computational complexity is lower and the fact that it does not require random sampling, KNNCLUST provides less accurate results than KSEM, but better than AP and NPSEM. It should be noticed that KSEM could not discriminate the classes *Grapes untrained* and *Vineyard untrained*, hence the 0% CSCCR obtained for the latter. This can be explained by the high similarity of these two classes in terms of spectral signatures due to close vegetation species.

Figure 6 shows instances of clustering maps provided by the four methods using the values of k specified as above. Visually, the clustering map obtained with KSEM is closer to the ground truth map than those of the other methods, as confirmed by the corresponding accuracy indices. Also note that two subclasses of *Celery* were identified by KSEM, though not clearly visible on Figure 6-(a). However, as said above, none of the methods was able to clearly discriminate the *Grapes untrained* and *Vineyard untrained* classes, except KNNCLUST, though quite marginally.

TABLE I

MEAN AND STANDARD DEVIATION (20 RUNS) OF CLASS SPECIFIC, AVERAGE, OVERALL ACCURACIES (IN PERCENT), CLUSTER PURITY AND NORMALIZED MUTUAL INFORMATION, NUMBER OF ITERATIONS AND EXECUTION TIME OF CLUSTERING METHODS FOR THE AVIRIS *Salinas* HYPERSPECTRAL DATA SET (AP: $\xi = 2.0$; KNNCLUST: $k = 50$; NPSEM: $k = 20, C_{\max} = 100$; KSEM: $k = 40$).

		Unsupervised classifier			
		AP	KNNCLUST	NPSEM	KSEM
	Total # clusters - min	35	27	14	20
	Total # clusters - med	39	32	18	22
	Total # clusters - max	43	37	23	26
	min. # clusters in GT	32	20	14	16
	med. # clusters in GT	36	25	18	18
	max. # clusters in GT	41	30	23	20
Classes	Brocc. gr. wds 1	96.78±1.97	**98.53±0.21**	68.06±45.73	93.34±21.97
	Brocc. gr. wds 2	49.01±9.68	98.07±7.12	83.34±21.68	**99.67±0.01**
	Fallow	46.07±3.13	51.92±11.50	36.56±23.39	**75.64±22.71**
	Fallow rgh pl.	67.32±13.88	63.69±28.97	74.43±39.87	**99.19±0.05**
	Fallow smooth	82.61±4.81	65.94±23.87	67.64±21.60	**91.37±5.66**
	Stubble	47.20±3.65	92.61±14.62	94.32±13.75	**99.73±0.02**
	Celery	55.00±3.67	74.50±19.52	57.24±21.17	**77.51±18.76**
	Grapes untrained	27.81±4.56	88.68±19.65	36.62±11.25	**99.57±0.02**
	Soil vin. devel.	56.52±8.60	**99.14±0.99**	60.52±16.79	95.65±9.32
	Corn sen. g. wds	55.80±0.79	54.78±13.23	60.31±16.30	**63.26±1.14**
	Lett. rom. 4 wks	**60.73±6.00**	60.15±28.61	59.00±39.18	47.07±39.31
	Lett. rom. 5 wks	53.77±5.04	87.18±23.31	55.34±32.12	**100**
	Lett. rom. 6 wks	**91.71±13.34**	74.26±43.99	55.58±49.00	88.92±30.41
	Lett. rom. 7 wks	75.71±13.69	78.65±19.68	76.18±32.92	**88.94±3.54**
	Vineyard untrained	**33.27±6.77**	41.77±47.51	**46.51±12.49**	0
	Vineyard ver. tr.	40.25±1.77	96.35±12.39	60.89±38.27	**99.22±0.05**
	ACCR	58.72±2.99	76.64±4.43	62.03±5.17	**82.44±3.08**
	OCCR	49.18±3.52	77.98±5.50	57.62±4.44	**79.17±2.11**
	Cluster purity	0.49±0.04	0.87±0.04	0.63±0.05	**0.94±0.02**
	NMI	0.69±0.01	0.88±0.01	0.70±0.03	**0.89±0.01**
	Iterations	251±56	23±5	74±23	107±15
	Exec. time (s)	1722±403	615±106	255±52	1363±176



(a) (b)

Fig. 4. *Salinas* hyperspectral data set.(a): Color composite image (bands 30, 20, 10); (b): Ground truth.

2) ROSIS - Pavia University: The *Pavia University* HSI belongs to a set of hyperspectral images acquired by the ROSIS instrument operated by the German Aerospace Agency (DLR) on July 8, 2002 in the framework of the European HySens Project. ROSIS provides 103 spectral bands ranging from 430 to 850 nm, at 1.3 m ground resolution. The *Pavia University* scene has a spatial size of 610×340 pixels. Nine classes are reported in the ground truth map visible in Figure 7 jointly with a color composite image. Table II shows the clustering maps and performance indices of the four compared methods. KNNCLUST, NPSEM and KSEM were run using $k = 60$, $k = 17$ and $k = 30$, respectively. These values were selected since because they provide the best average accuracies for each method. A median number of 19 clusters were found by KSEM, with a median number of 16 clusters within the nine ground truth classes, giving an average ACCR of 63.20% over 10 runs, again better than the other compared methods. NPSEM, which is faster than the other methods, does not provide satisfactory results in this experiment, though slightly superior to AP. Here again, Cluster Purity and NMI indices are in accordance with ACCR and OCCR.

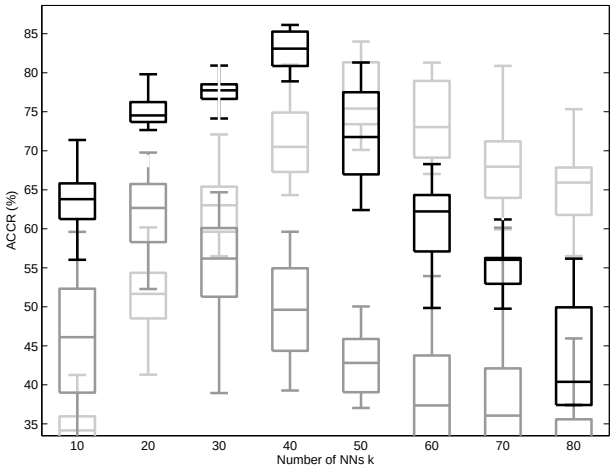


Fig. 5. Box plots of Average Correct Classification rate (ACCR) for the *Salinas* HSI as a function of the number of nearest neighbors k. Light gray: KNNCLUST; Gray: NPSEM; Black: KSEM.

Figure 8 displays typical clustering maps issued from this experiment, and gives the corresponding correct classification rates.

3) AVIRIS - Hekla: In the last experiment, we used a HSI collected on 17 June 1991 by AVIRIS over the Hekla volcano in Iceland. The image has 560×600 pixels, with 157 spectral bands only due to a sensor malfunctioning. The ground resolution is 20 m. Figure 9 shows a color composite image as well as the ground truth patches used for the clustering assessment, which comprises twelve land-cover classes.

Table III displays as above the performance indices of the four methods averaged over 10 runs, using $k = 10$ for KSEM, $k = 30$ for KNNCLUST and $k = 7$ for NPSEM, which were chosen as optimal regarding the average ACCR. In this experiment, AP and NPSEM still provide poor results, whilst KNNCLUST and KSEM provide similar results, with a slightly higher (but non significant) ACCR for KNNCLUST. Despite the higher computational burden of KSEM, one can see from this experiment that a random sampling approach can perform as better as a deterministic approach, by using a reduced kNN

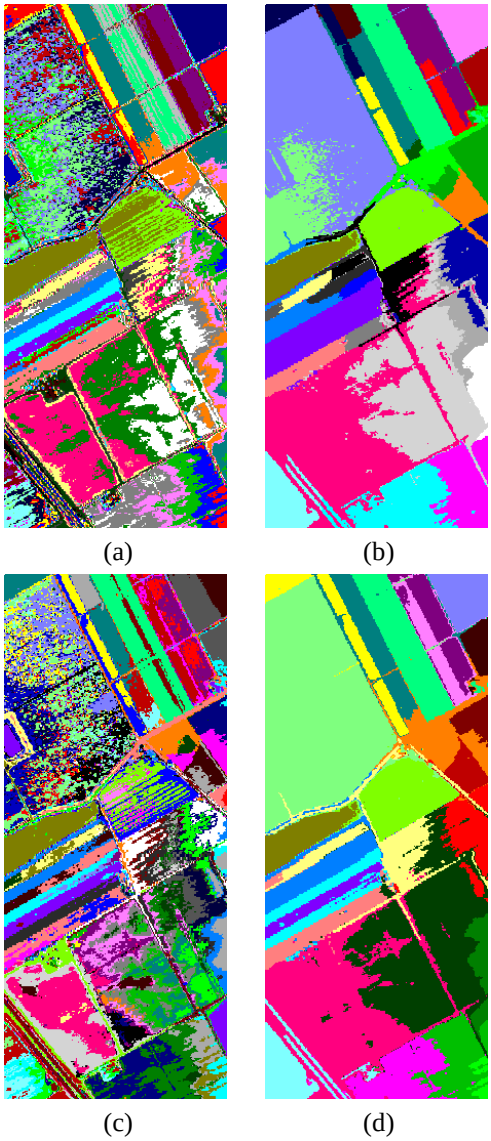


Fig. 6. Unsupervised classification results for the AVIRIS *Salinas* hyperspectral data set. (a): AP (OCCR: 54.34%, ACCR: 62.51%); (b): KNNCLUST (OCCR: 72.81%, ACCR: 70.23%); (c): NPSEM (OCCR: 45.43%, ACCR: 52.05%); (d): KSEM (OCCR: 78.44%, ACCR: 83.06%).

graph. It is also noticeable that the median number of clusters found by KSEM within the ground truth pixels (25) is closer to the true number of known classes than any of the other methods.

Examples of clustering maps provided by the four clustering methods are shown in Figure 10, with associated OCCR and ACCR values. From these examples, a comparison of the behavior of KNNCLUST and KSEM results for the specific class *Andesite lava 1991 I* (large region at top right, in yellow on Figure 10-(d)) highlights the limitation of the deterministic probability update rule of KNNCLUST, which tends to relax the labeling from seed pixels or regions located far apart, without possibility to merge these labels into a single one, hence providing over clustering. Yet, performing random label assignments according to conditional local distributions allows to gain in clustering robustness thanks to the fact that label propagation from a seed region to another is made possible.

TABLE II
MEAN AND STANDARD DEVIATION (10 RUNS) OF CLASS SPECIFIC, AVERAGE, OVERALL ACCURACIES (IN PERCENT), CLUSTER PURITY AND NORMALIZED MUTUAL INFORMATION, NUMBER OF ITERATIONS AND EXECUTION TIME OF CLUSTERING METHODS FOR THE ROSIS *Pavia university* HYPERSPECTRAL DATA SET (AP: $\xi = 3.0$; KNNCLUST: $k = 60$; NPSEM: $k = 17, C_{\max} = 100$; KSEM: $k = 30$).

		Unsupervised classifier			
		AP	KNNCLUST	NPSEM	KSEM
	Total # clusters - min	29	21	19	17
	Total # clusters - med	33	25	23	19
	Total # clusters - max	36	29	27	24
	min. # clusters in GT	29	15	19	14
	med. # clusters in GT	33	20	23	16
	max. # clusters in GT	36	23	27	20
Classes	Asphalt	**33.57±3.87**	28.10±8.09	29.11±4.58	33.20±10.51
	Meadows	17.36±4.19	47.84±16.68	23.35±6.43	**55.74±4.65**
	Gravel	41.97±10.84	**54.15±25.01**	41.97±17.00	50.84±32.37
	Trees	28.86±3.62	21.69±10.67	34.84±9.63	**52.95±15.56**
	(Painted) metal sheets	46.46±8.15	89.10±31.31	84.46±20.75	**98.79±0.58**
	Bare soil	16.22±2.34	**91.25±13.81**	37.51±2.74	80.90±15.34
	Bitumen	75.15±18.22	**99.83±0.06**	84.31±18.93	79.86±42.05
	Self-blocking bricks	**50.65±10.31**	49.87±6.75	42.77±5.78	48.07±5.62
	Shadow	**97.18±1.33**	37.64±18.62	73.58±27.58	68.48±25.11
	ACCR	45.27±2.48	57.72±4.88	50.21±3.97	**63.20±4.92**
	OCCR	29.39±2.37	50.92±7.37	34.41±2.72	**56.40±3.33**
	Cluster purity	0.30±0.02	0.54±0.08	0.35±0.03	**0.59±0.04**
	NMI	0.48±0.00	0.61±0.04	0.52±0.03	**0.63±0.02**
	Iterations	251±61	31±4	84±18	93±7
	Exec. time (s)	1764±512	1733±208	546±63	2323±185

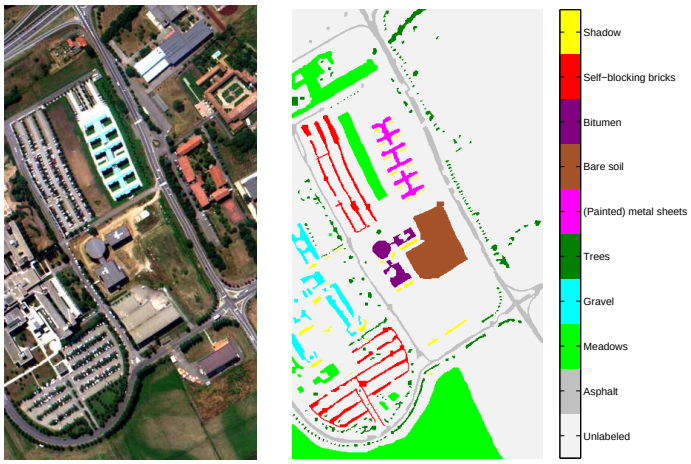


Fig. 7. *Pavia university* hyperspectral data set.(a): Color composite image (bands 60, 33, 9); (b): Ground truth.

IV. CONCLUSION

In this paper, we proposed a new unsupervised clustering method, named KSEM, which is based on iteratively sampling label states via pseudo-posterior label distributions estimated at the objects' local level. Contrarily to many clustering methods, KSEM is fully unsupervised since it has the ability to provide an estimate of the number of clusters in the data, starting from one distinct cluster label by object. The local posterior distributions account for the number of similar labels among the kNNs of each object, and class-conditional differential entropies computed thanks to the Kozachenko-Leonenko estimator are used to elaborate a stopping criterion. A probability reinforcement rule is set up to accelerate the convergence to a stable partitioning of the objects. The method is compared with three other fully unsupervised clustering methods for purposes of pixel clustering in hyperspectral images. A specific processing is set up in KSEM (and also adapted to KNNCLUST and NPSEM) to make the prior kNN search procedure tractable for (possibly large) image data sets. The results show the

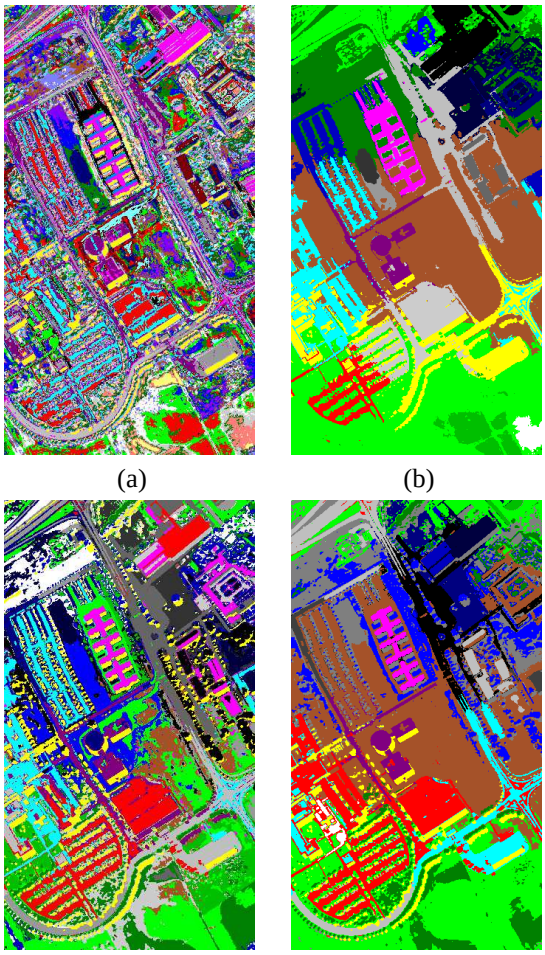


Fig. 8. Unsupervised classification results for the ROSIS *Pavia University* hyperspectral data set. (a): AP (OCCR: 28.98%, ACCR: 41.91%); (b): KNNCLUST (OCCR: 55.97%, ACCR: 62.82%); (c): NPSEM (OCCR: 39.89%, ACCR: 54.12%); (d): KSEM (OCCR: 58.04%, ACCR: 64.68%).

efficiency of the proposed approach in retrieving coherent clusters with respect to available ground truth data.

ACKNOWLEDGEMENTS

Authors wish to thank Prof. J. A. Benediktsson from the University of Iceland, Reykjavik, Iceland, for providing the AVIRIS *Hekla* data set, and to Prof. P. Gamba from the University of Pavia, Italy, for providing the ROSIS *Pavia University* data set.

TABLE III

MEAN AND STANDARD DEVIATION (10 RUNS) OF CLASS SPECIFIC, AVERAGE, OVERALL ACCURACIES (IN PERCENT), CLUSTER PURITY AND NORMALIZED MUTUAL INFORMATION, NUMBER OF ITERATIONS AND EXECUTION TIME OF CLUSTERING METHODS FOR THE AVIRIS *Hekla* HYPERSPECTRAL DATA SET (AP: $\xi = 1.0$; KNNCLUST: $k = 30$; NPSEM: $k = 7, C_{\max} = 100$; KSEM: $k = 10$).

		Unsupervised classifier			
		AP	KNNCLUST	NPSEM	KSEM
	Total # clusters - min	33	53	30	37
	Total # clusters - med	35	57	35	42
	Total # clusters - max	38	66	39	46
	min. # clusters in GT	33	25	30	21
	med. # clusters in GT	35	29	34	25
	max. # clusters in GT	37	34	39	30
Classes	Andesite lava 1970	82.89±18.21	97.34±5.55	96.08±3.50	**98.98±0.57**
	Andesite lava 1980 I	53.84±17.87	63.90±5.63	77.73±19.84	**80.75±16.46**
	Andesite lava 1980 II	81.86±7.63	**99.96±0.06**	94.88±11.13	99.95±0.04
	Andesite lava 1991 I	58.83±7.76	84.91±19.49	14.65±4.77	**96.75±9.64**
	Andesite lava 1991 II	27.02±7.72	**58.22±15.56**	47.68±12.37	37.12±19.78
	Andesite lava moss cover	62.06±10.17	80.96±12.99	81.08±14.36	**95.81±9.42**
	Hyaloclastite formation	37.28±8.30	**84.02±12.35**	59.23±20.37	77.75±12.09
	Lava tephra covered	73.46±17.67	**98.89±1.13**	88.76±24.24	79.77±42.05
	Rhyolite	29.23±3.38	76.58±15.75	76.56±35.09	**100**
	Scoria	16.98±6.09	**43.35±15.11**	38.15±18.06	25.40±16.19
	Firn-glacier ice	35.68±5.99	**75.02±17.92**	56.90±16.69	66.90±0.76
	Snow	61.64±12.29	**72.40±16.17**	52.01±10.42	54.36±2.92
	ACCR	51.73±3.79	**77.96±3.26**	65.31±5.29	76.13±4.06
	OCCR	57.01±3.21	81.63±6.12	57.70±5.11	**83.27±3.31**
	Cluster purity	0.59±0.03	0.82±0.06	0.59±0.05	**0.88±0.03**
	NMI	0.68±0.02	**0.90±0.02**	0.69±0.05	0.89±0.02
	Iterations	203±22	26±2	127±41	201±14
	Exec. time (s)	1440±264	1878±92	933±242	7413±562

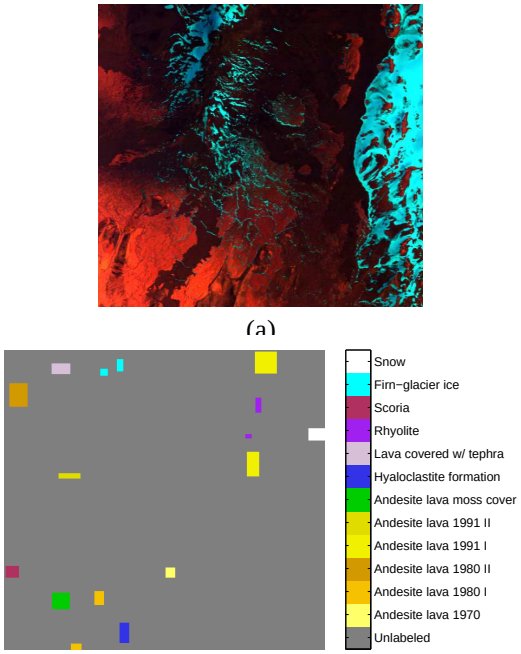


Fig. 9. *Hekla* hyperspectral data set.(a): Color composite image (bands 29, 15, 9); (b): Ground truth.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[2] A. K. Jain, "Data clustering: 50 years beyond k-means." *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[3] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised clustering by seeding." in *ICML*, C. Sammut and A. G. Hoffmann, Eds. Morgan Kaufmann, 2002, pp. 27–34.

[4] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[5] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[6] J. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, pp. 236–244, 1963.

[7] P. Sneath and R. Sokal, *Numerical Taxonomy. The Principles and Practice of Numerical Classification*. London, UK: Freeman, 1973.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd international conference on Knowledge Discovery and Data mining KDD'96*, 1996, pp. 226–231.

[9] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99)*, Philadelphia, PA, 1999, pp. 49–60.

[10] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition." *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

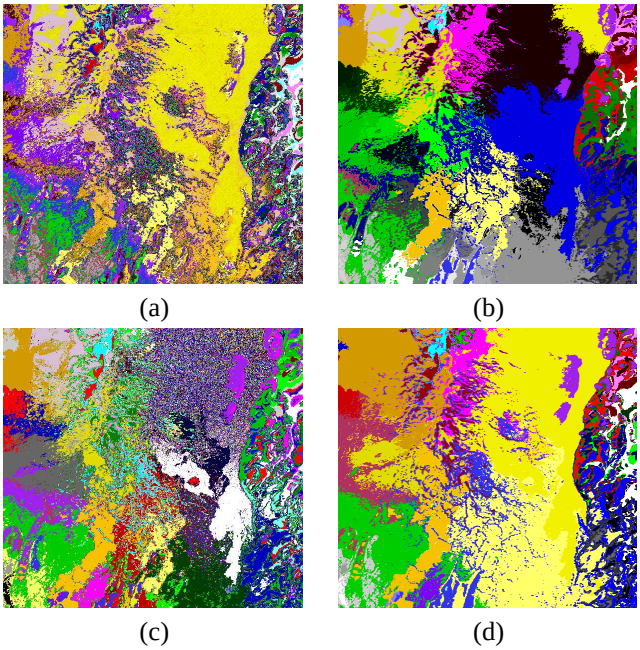


Fig. 10. Unsupervised classification results for the AVIRIS *Hekla* hyperspectral data set. (a): AP (OCCR: 55.56%, ACCR: 50.26%); (b): KNNCLUST (OCCR: 72.00%, ACCR: 72.07%); (c): NPSEM (OCCR: 52.99%, ACCR: 58.87%); (d): KSEM (OCCR: 80.96%, ACCR: 72.60%).

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977, series B.

[12] G. Celeux and J. Diebolt, "A probabilistic teacher algorithm for iterative maximum likelihood estimation," in *Classification and related methods of data analysis*, H. H. Bock, Ed. North-Holland: Elsevier, 1988, pp. 617–623.

[13] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," *Computational Statistics & Data Analysis*, vol. 14, no. 3, pp. 315–332, Oct. 1992.

[14] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems (NIPS)*, S. A. Solla, T. K. Leen, and K. R. Müller, Eds. MIT Press, 2000, pp. 554–560.

[15] C. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems," *The Annals of Statistics*, vol. 2, pp. 1152–1174, 1974.

[16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[17] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[18] I. S. Dhillon, S. Mallela, and R. Kumar, "A divisive information theoretic feature clustering algorithm for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1265–1287, 2003.

[19] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[20] L. Faivishevsky and J. Goldberger, "A Nonparametric Information Theoretic Clustering Algorithm," in *International Conference on Machine Learning*, Haifa, Israel, 2010.

[21] M. Wang and F. Sha, "Information theoretical clustering via semidefinite programming." in *International Conference on Artificial Intelligence and Statistics*, ser. JMLR Proceedings, G. J. Gordon, D. B. Dunson, and M. Dudk, Eds., vol. 15. Ft. Lauderdale, FL, USA: JMLR.org, 2011, pp. 761–769.

[22] A. C. Müller, S. Nowozin, and C. H. Lampert, "Information theoretic clustering using minimum spanning trees." in *DAGM/OAGM Symposium*, ser. Lecture Notes in Computer Science, A. Pinz, T. Pock, H. Bischof, and F. Leberl, Eds., vol. 7476. Springer, 2012, pp. 205–215.

[23] G. Ver Steeg, A. Galstyan, F. Sha, and S. DeDeo, "Demystifying information-theoretic clustering," in *International Conference on Machine Learning*, Beijing, China, 2014.

[24] M. Sugiyama, G. Niu, M. Yamada, M. Kimura, and H. Hachiya, "Information-maximization clustering based on squared-loss mutual information." *Neural Computation*, vol. 26, no. 1, pp. 84–131, 2014.

[25] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, February 2007.

[26] J. Kittler and J. Illingworth, "Relaxation labelling algorithms - a review." *Image and Vision Computing*, vol. 3, no. 4, pp. 206–216, 1985.

[27] G. Celeux and J. Diebolt, "The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem." *Comput. Statist. Quarter.*, vol. 2, pp. 73–82, 1985.

[28] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1979.

[29] J. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, September 1973.

[30] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD'96)*, 1996, pp. 103–114.

[31] A. Lorette, X. Descombes, and J. Zerubia, "Fully unsupervised fuzzy clustering with entropy criterion." in *Proc. 15th International Conference on Pattern Recognition (ICPR 2000)*, vol. 3, Barcelona, Spain, Sept. 2000, pp. 986–989.

[32] T. S. Ferguson, "Bayesian density estimation by mixtures of normal distributions," in *Recent Advances in Statistics*, M. Rizvi, J. Rustagi, and D. Siegmund, Eds. New York: Academic Press, 1983, pp. 287–302.

[33] D. Aldous, "Exchangeability and related topics," in *École d'été de probabilités de Saint-Flour, XIII—1983*. Berlin: Springer, 1985, pp. 1–198.

[34] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[35] T. N. Tran, R. Wehrens, and L. M. C. Buydens, "Knn-kernel density-based clustering for high-dimensional multivariate data," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 513–525, Nov. 2006.

[36] C. Robert and G. Casella, "A short history of Markov Chain Monte Carlo: Subjective recollections from incomplete data," *Statistical Science*, vol. 26, no. 1, pp. 102–115, 2011.

[37] A. Samé, G. Govaert, and C. Ambroise, "A mixture model-based on-line CEM algorithm." in *IDA*, ser. Lecture Notes in Computer Science, A. F. Famili, J. N. Kok, J. M. Pea, A. Siebes, and A. J. Feelders, Eds., vol. 3646. Springer, 2005, pp. 373–384.

[38] G. Bougenière, C. Cariou, K. Chehdi, and A. Gay, "Unsupervised non parametric data clustering by means of Bayesian inference and information theory." in *SIGMAP*, S. M. M. de Faria and P. A. A. Assuno, Eds. INSTICC Press, 2007, pp. 101–108.

[39] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *IEEE Conference on Decision and Control*, vol. 17, Jan 1978, pp. 761–766.

[40] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 282–289.

[41] G. Bougenière, C. Cariou, K. Chehdi, and A. Gay, "Non parametric stochastic expectation maximization for data clustering," in *E-business and Telecommunications*, ser. Communications in Computer and Information Science, J. Filipe and M. S. Obaidat, Eds. Springer Berlin Heidelberg, 2009, vol. 23, pp. 293–303.

[42] M. Jardino, "Unsupervised non-hierarchical entropy-based clustering," in *Data Analysis, Classification, and Related Methods*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, H. A. L. Kiers, J.-P. Rasson, P. J. F. Groenen, and M. Schader, Eds., vol. I. Springer, 2000, pp. 29–34.

[43] R. Kneser and H. Ney, "Improved clustering techniques for class-based statistical language modelling," in *Proc. Eurospeech 93*, 1993, pp. 973–976.

[44] L. F. Kozachenko and N. N. Leonenko, "Sample estimate of the entropy of a random vector," *Problemy Peredachi Informatsii*, vol. 23, no. 2, pp. 9–16, 1987, [in Russian].

[45] M. N. Goria, N. N. Leonenko, V. V. Mergel, and P. L. N. Inverardi, "A new class of random vector entropy estimators and its applications in testing statistical hypotheses," *Journal of Nonparametric Statistics*, vol. 17, no. 3, pp. 277–297, 2005.

[46] N. Leonenko, L. Pronzato, and V. Savani, "Estimation of entropies and divergences via nearest neighbors," *Tatra Mountains Mathematical Publications*, vol. 39, pp. 265–273, 2008.

[47] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, “Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2973–2987, Aug. 2009.
[48] K. Chehdi, M. Soltani, and C. Cariou, “Pixel classification of large size hyperspectral images by affinity propagation,” *Journal of Applied Remote Sensing*, vol. 8, no. 1, August 2014.
[49] H. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[50] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge University Press, 2008.
[51] http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

Claude Cariou received the Ph.D. degree in Electronics from the University of Brest, France, in 1991. Since 1992, he has been with the Engineering School of Applied Sciences and Technology (ENSSAT), where he is currently with the Institute of Electronics and Telecommunications of Rennes, France. His research interests include image analysis, pattern recognition, unsupervised classification, texture modeling and segmentation, image registration and feature extraction/selection, mostly dedicated to multispectral and hyperspectral imagery.

Kacem Chehdi received the Ph.D. and the ”Habilitation à diriger des Recherches” degrees in Signal Processing and Telecommunications from the University of Rennes 1, France, in 1986 and 1992, respectively. From 1986 to 1992, he was an Assistant Professor at the University of Rennes 1. Since 1993, he has been a Professor of signal and image processing at the same institution. From 1998 to 2003, he was the Head of the laboratory ”Analysis Systems of Information Processing”. Since 2004, he is the Head of the TSI2M Laboratory (Signal and Multicomponent/Multimodal Image Processing). His research activities concern adaptive processing at every level in the pattern recognition chain by vision. In the framework of blind restoration and blind filtering, his main interests are the identification of the physical nature of image degradations and the development of adaptive algorithms. In the segmentation and registration topics, his research concerns the development of unsupervised, cooperative and adaptive systems. The main application under current investigation are multispectral and hyperspectral image processing and analysis.